Explicit window adaptation algorithm over TCP wireless networks

H.-J. Byun and J.-T. Lim

Abstract: A feedback-based congestion control algorithm is described which improves transmission control protocol (TCP) performance over wireless networks. An explicit feedback scheme is proposed for the fair sharing of the bandwidth by modifying the receiver’s advertised window in TCP acknowledgments returning to the source. Using the feedback information, the proposed algorithm adapts the window size to avoid the congestion and sees the packet loss only due to the wireless link error. Based on the asymptotic analysis, it is shown that the proposed algorithm guarantees the fair sharing and improves wireless TCP performance. The effectiveness of the proposed algorithm is demonstrated by simulations.

1 Introduction

Transmission control protocol (TCP) is designed for wired networks where the channel error rates are very low and congestion is the primary cause of packet loss. TCP assumes that every packet loss is an indication of the network congestion and reduces the transmission rate. However, when a wireless link forms a part of a network, the packet losses due to the wireless link error are often more significant than the ones due to the congestion. Since TCP has no mechanism to differentiate these losses from congestion, it treats all losses as congestive by reducing its transmission rate, and therefore TCP shows greatly degraded performance over wireless networks. Many algorithms to improve wireless TCP performance have been proposed [1–15]. In [1, 2], the source is informed of the reason for a packet loss explicitly. The protocols in [3–7] decide whether packet losses are likely to be due to congestion or wireless link errors using the network information. The protocols proposed in [8–10] eliminate the packet loss due to the buffer overflows, so the source only sees the wireless losses. The algorithms in [11, 12] estimate the packet rate of the connection by monitoring ACK reception rate. In [13–15], the explicit window adaptation algorithms have been proposed. These algorithms suggest an explicit congestion control scheme of the window size of TCP connections by signalling the window update to the TCP source. This is achieved by explicitly controlling the feedback value from the network elements. Specifically, once the window size is computed, network elements signal the window update to the TCP source by modifying the receiver’s advertised window (AWND) field carried by TCP ACKs. However, they achieve fairness only in the homogeneous network, where all the connections have the same round-trip time (RTT). In addition, there is far less result on methods to provide fair sharing of the bandwidth and high link utilisation over wireless TCP networks with high variance in their RTT distribution.

In this paper, we propose an algorithm to improve the TCP performance and avoid packet losses in the wireless networks based on the explicit window adaptation scheme [13–15].

2 Proposed algorithm

We consider a single bottleneck link where each connection has a different propagation delay. In order to define the network under consideration, we assume the following:

- round-trip time of TCP connection i is RTTi
- outgoing link of the base station is the bottleneck link of the network
- sending host updates its window size every round-trip time.

First, we introduce link price pi (0 ≤ pi ≤ 1). For link l, link price pi is updated every τi according to

\[ p_i = \begin{cases} p_{\text{max}} \frac{q_i}{q^*} & \text{if } q_i \leq q^* \\ 1 & \text{else} \end{cases} \]  

(1)

where \( q_i \) is the queue length of link l, \( p_{\text{max}} \) (0 < \( p_{\text{max}} \) ≤ 1) is the maximum link price, \( q^* \) is the queue threshold and \( \tau_i \) is a constant.

The proposed algorithm sends the explicit feedback computed by the router to TCP sources to adjust their window sizes. That is, using the link price, the feedback value \( FV \) is computed by router l as

\[ FV_l = \left[ \tau_l (p_{\text{max}} - p_i) \right]^+ \]  

(2)

where \( \tau_l \) is a control gain to be chosen and \([ \cdot ]^+ = \max(0, \cdot)\). The feedback value is based on the difference between \( p_{\text{max}} \) and \( p_i \). This makes the system keep the queue length smaller than the queue threshold irrespective of the number of active connections.

The feedback value is computed every \( \tau_i \) and carried by returning TCP ACKs in the AWND field. If the current value in the receiver’s advertised window exceeds the feedback value computed by the router, the receiver’s advertised window is reduced to the computed feedback.
value. Ultimately, the AWND field will contain the optimal feedback value along the path. Since the router uses the queue length to compute the feedback value for the active connections, the proposed scheme does not require maintaining per-connection state at the routers. Therefore all connections routed through the bottleneck link will receive the same feedback value.

When the feedback value reaches the sender, the sender updates its current window size using the received feedback value. We suppose that link l is the bottleneck link in the network. To guarantee the fair sharing of the available bandwidth and match the aggregate window size of all active TCP connections to the network pipe, the window size for connection i, $w_i$, is set to the received feedback value multiplied by RTT$_i$:

$$w_i = \max(FV_i \cdot RTT_i, MSS)$$  \hspace{1cm} (3)

where MSS is the maximum segment size. Since setting the window size smaller than MSS can lead to starvation and deadlocks, a minimum window size of MSS is enforced.

3 Analysis

3.1 Network modelling

For the analysis, we suppose that the computed window size is always greater than MSS. That is, the window size is determined by the feedback value multiplied by RTT. Let $M$ be the number of active connections routed through the bottleneck link and let $\tau_i$ be a minimum RTT of connection i (1 $\leq i \leq M$), which is obtained when the network is not congested. In addition, $\tau_f$ and $\tau_b$ are the forward/backward delay of connection i. The capacity of outgoing bottleneck link l is denoted by $C_l$ [packet/s]. Assume that $\tau_1 < \tau_2 < \cdots < \tau_{M-1} < \tau_M$.

Let $\Delta_i = \left\lceil \frac{\tau_i}{\tau_f} \right\rceil$ where $[x]$ is the smallest integer larger than x. Provided that the round-trip propagation time is dominant over the waiting time of a packet at the router, the system can be represented by a discrete-time model, where $\tau_s$ is the duration of a normalised time slot. Let $w_f(n)$ denote the window size of the sending host i at time n. From the assumption that the sending host updates its window size every $\tau_s$, (3) can be rewritten as follows:

$$w_f(n + 1) = \left\{ \begin{array}{ll}
[\frac{[x_i(p_{\text{max}} - p(n - d_b))]^{+} \cdot \tau_s}{w_f(n) - \text{else}}
\end{array} \right. \hspace{1cm} (4)

where $d_b = \lceil \frac{\tau_b}{\tau_f} \rceil$. Let $e(n)$ denote the stationary process of packet loss ratio at time n on the wireless link. Then, the dynamics of the bottleneck queue l is described by the following difference equation:

$$q_l(n + 1) = [q_l(n) + (R(n) - C_l(1 - e(n))\tau_s)]^{+} \hspace{1cm} (5)

where $R(n) = \sum_{i=1}^{M} w_f(n + 1 - d_f) / \tau_i$ and $d_f = \lceil \frac{\tau_f}{\tau_s} \rceil$.

Originally, the window-based congestion control mechanism allows the sending host to send $w_i$ per its RTT. However, since each sender updates the window size independently every RTT, it is difficult to analyse the asymptotic stability in the heterogeneous network. In this Section, we approximate the network model based on the sense of average. Specifically, we assume that connection i sends $w_i / \Delta_i$ on average and updates the window size every normalised time slot. Thus, the window dynamics of (4) and queue dynamics of (5) are rewritten by the following equation:

$$\dot{w}_i(n + 1) = [x_i(p_{\text{max}} - p(n - d_b))]^{+} \cdot \tau_s \hspace{1cm} (6)

$$q_l(n + 1) = [q_l(n) + \bar{W}(n) - C_l(1 - e(n))\tau_s]^{+} \hspace{1cm} (7)

where $\bar{W}(n) = \sum_{i=1}^{M} \dot{w}_i(n + 1 - d_i)$. Substituting (1) and (6) to (7), we derive the closed-loop dynamics for the network model.

$$q_l(n + 1) = \left[ q_l(n) + \frac{p_{\text{max}}}{q_s} \sum_{i=1}^{M} (q_s - q_l(n - \Delta_i))\tau_s - C_l(1 - e(n))\tau_s \right]^{+} \hspace{1cm} (8)

Let

$$x(n) = \frac{q_l(n)}{q_e} \hspace{1cm} (9)

and

$$k_1 = \frac{\epsilon_a}{\epsilon}, \hspace{0.5cm} k_2 = \frac{\epsilon_b}{\epsilon}

where

$$\epsilon_a = x_1 \frac{p_{\text{max}}}{q_e} \tau_s, \hspace{0.5cm} \epsilon_b = \frac{C_l\tau_s}{q_e}, \hspace{0.5cm} \epsilon = \min(\epsilon_a, \epsilon_b)

and $q_e$ is the total queue capacity. Then (8) is written as

$$x(n + 1) = x(n) + \epsilon \Phi(x(n), x(n - 1), \ldots, x(n - D), e(n)) \hspace{1cm} (10)

where

$$\Phi(x(n), x(n - 1), \ldots, x(n - D), e(n)) \triangleq

- k_1 \sum_{i=1}^{M} \left[ \frac{x(n - \Delta_i) - q_s}{q_e} + k_2(1 - e(n)) \right]

and $D = \max(\Delta_i)$. Since $q_e \gg 1$, without losing much generality, we assume that $\epsilon \ll 1$. Note that since $\Phi$ can take arbitrarily large values, but with negligibly small probabilities, (10) represents a slow-in-the-average Markov walk process [16, 17]. Let $y(n)$ denote the averaged value of $x(n)$. Applying the asymptotic theory for such processes [16, 17], we obtain the following asymptotic approximation:

$$y(n + 1) = y(n) - \epsilon \left( k_1 \sum_{i=1}^{M} \left[ x(n - \Delta_i) - q_s \right] + k_2(1 - e) \right) \hspace{1cm} (11)

where $e = E[e(n)]$.

3.2 Steady state and fairness

Let $q_a$ and $\dot{w}_a$ denote the steady-state solution of $q_l(n)$ and $\dot{w}_a(n)$. Note that in the neighbourhood of the steady state, we ignore the saturation nonlinearity. Thus, from (6)–(9) and (11), if $x_i > \frac{C_l(1 - e)}{M p_{\text{max}}}$, the steady state is obtained from

$$q_a = q^* - \frac{C_l(1 - e) q^*}{x_i M} \hspace{1cm} (12)

$$\dot{w}_a = \frac{C_l\tau_s}{M} (1 - e) \hspace{1cm} (13)

Note that $q_a$ cannot be greater than $q^*$. In addition, since $x_i > \frac{C_l(1 - e)}{M p_{\text{max}}}$, $q_a$ can be stabilised at a certain value smaller
than $q^*$ but greater than zero. Since $\dot{w}_i$ is the steady-state solution of (6), which represents the approximated window dynamic at every normalised time slot, we obtain the window size of connection $i$ at every RTT by multiplying $\Delta$, to (13). That is $w_i = \frac{C_i(1-e)}{M} (1-e)$ for all $i$ and the transmission rate $\lambda_i$ of connection $i$ at the steady state is the same as $\frac{C_i}{M} (1-e)$. Here, the average link capacity is $C_i(1-e)$. Therefore, if $x_t$ is chosen such that $x_t > \frac{C_i(1-e)}{M \cdot p_{\text{max}}}$, the system guarantees the fair sharing of the bottleneck link bandwidth and high link utilisation.

Since TCP connection adapts its window size using the feedback information to avoid the packet losses due to the buffer overflow, the proposed algorithm only sees the wireless losses and treats it as an indication of the wireless link error when a packet loss occurs. Accordingly, a TCP sender does not reduce the window size as a response to the packet loss.

Now we investigate the asymptotic stability of the steady state of (11) shown in (12). Let $\dot{y}(n) = y(n) - y_s$, where $y_s = \frac{2}{M}$. Then

$$\dot{y}(n + 1) = \dot{y}(n) - \epsilon_a \sum_{d=0}^{D} l_d \dot{y}(n - d)$$

(14)

Let $Y(n)$ be the state vector with respect to the queue dynamics of the bottleneck link that is represented by

$$Y(n) = [\dot{y}(n) \quad \dot{y}(n-1) \ldots \quad \dot{y}(n-D)]^T$$

Then, (14) can be written by the following state equation:

$$\dot{Y}(n + 1) = AY(n)$$

(15)

where

$$A = \begin{bmatrix}
1 - \epsilon_a l_0 & -\epsilon_a l_1 & \ldots & -\epsilon_a l_{D-1} & -\epsilon_a l_D \\
1 & 0 & \ldots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \ldots & 1 & 0
\end{bmatrix}$$

For the bottleneck link, the characteristic polynomial of $A$ of (15) is obtained as follows:

$$P(z) = z^{D+1} - z^D + \epsilon_a \sum_{d=0}^{D} l_d z^{D-d}$$

(16)

From [18], $P(z)$ has all zeros within the unit circle and then, the steady state is asymptotically stable if the control gain $x_t$ satisfies the following relation:

$$0 < x_t < \frac{2}{M \cdot p_{\text{max}} \cdot \tau_s} \left( \frac{\pi}{4D + 2} \right)$$

(17)

Accordingly, all eigenvalues of $A$ are located within the unit circle. Therefore, for the overall system, the steady state is also asymptotically stable. Finally, from the conditions of the high utilisation and of the stability, we conclude that if the control gain $x_t$ satisfies the following relation, the system guarantees the high utilisation and stability.

$$\frac{C_i(1-e)}{M \cdot p_{\text{max}}} < x_t < \frac{2}{M \cdot p_{\text{max}} \cdot \tau_s} \sin \left( \frac{\pi}{4D + 2} \right)$$

(18)

Our proposed technique can be applied to ad hoc networks in cases where the ad hoc nodes support the feedback signalling method and the route-failure notification method. In ad hoc networks, compared to the wireless link error or the congestion, node mobility, especially mobility-induced network disconnection and reconnection events, has the most significant impact on TCP performance. Hence, the route-failure notification, such as ELF (explicit link failure notification) [19] or TCP-F (TCP-feedback) [20], is needed to provide the TCP sender with information about link and route failures so that it can avoid responding to the failures as if congestion occurred. However, our proposed algorithm considers only the packet losses due to the wireless link error and congestion. Therefore, our techniques can be applied to ad hoc networks and work well provided that the route-failure notification method is supported.

4 Simulation results

In this Section, we describe simulation results of the proposed algorithm. We compare our proposed algorithm with TCP Veno [5]. TCP Veno monitors the network congestion level and uses that information to decide whether packet losses are likely to be due to congestion or random bit error. Let us consider a simplified network model shown in Fig. 1. The link between R2 and R3 is assumed to be the wireless bottleneck link in this model and the wired network is a LAN speed network. We use the following network parameters: the wireless link capacity $C_i$ is 1500 [packet/s], $p_{\text{max}}$ is 1 and $q^*$ is 100 packets. The number of TCP connection $M$ is 3 and the minimum RTT, $\tau_s$, of each TCP connection is 10, 20 and 30 ms, respectively. We set $\tau_s = 10$ ms and choose $x_t = 600$ according to (18). The packet size is 512 bytes.

![Network model](image)

**Fig. 1** Network model

Figure 2 shows the results of TCP Veno [5] in terms of the window size, queue length, throughput and link utilisation for a wireless bit error rate (BER) of $1 \times 10^{-5}$, which corresponds to the packet error rate (PER) of 4%. As shown in Fig. 2, there is an excessive oscillation in the window size and queue length, though this algorithm employs the modified Reno that reduces the sending rate less aggressively when random loss is detected. In addition, the fair sharing of the bandwidth and the high link utilisation are not achieved for a considerable period.

Figure 3 shows the results of the proposed algorithm. In Fig. 3a, the window size obtained from the proposed algorithm shows superior behaviour, i.e. there is fast convergence without excessive oscillations. In addition, the window size of the connection with larger RTT is larger than that of a connection with smaller RTT. This is due to the proper adaptation of the window size in order to obtain fair bandwidth allocation. As shown in Fig. 3b, the queue length converges around the steady state smaller than $q^*$. The throughput has converged to the fair share and high link utilisation is achieved.

To show the fairness in the throughput, we measure the fairness index [21] $I(n)$ of the throughput of all connections, which is given by

$$I(n) = \frac{\left( \sum_{i=1}^{M} \lambda_i(n) \right)^2}{M \sum_{i=1}^{M} \lambda_i^2(n)}$$

where $\lambda_i(n)$ is the transmission rate of connection $i$ and is equal to the window size divided by its round-trip
time. The value of this index is between 0 and 1. If the throughput of each connection is exactly the same, the value of the index will be 1. The smaller the index, the larger the unfairness in throughput. Figure 4 shows the fairness index. Compared with TCP Veno, the fairness index of the proposed algorithm tends to
converge to 1, but TCP Veno has the smaller index.

Figure 5 shows the throughput averaged during 10 s of TCP Veno and the proposed algorithm for a bit error rate in the range $1 \times 10^{-7} \sim 5 \times 10^{-5}$. TCP Veno performs reasonably well when the bit error rate is small, but as the bit error rate increases its performance degrades more quickly than that of the proposed algorithm. In addition, we find that the the proposed algorithm guarantees the fair sharing of the bandwidth for a wide range of bit error rate.

5 Conclusions

In this paper, we propose an explicit window adaptation algorithm to improve TCP performance over wireless
networks. We introduce the link price to compute the feedback value and the computed feedback values are signalled to TCP sources by modifying the AWND field carried by TCP ACKs. From the simulation, we observe significant improvements in the performance over TCP wireless networks with the proposed algorithm.

6 Acknowledgments

This work was supported by HWRS Engineering Research Center, KAIST.

7 References